Anomaly Detection in IoT using Enhanced K-means, AGNES Clustering, and Echo State Networks

Shanshan Li*, Xiafei Chen
Office of Network and Information Technology, Henan University of Science and Technology, Luoyang 471000, China
E-mail: 15670383034@163.com
*Corresponding author

Keywords: abnormal behavior, K-means, AGNES, ENS, Internet of things

Received: May 17, 2025

In response to the poor performance of traditional Internet of Things (IoT) anomaly behavior detection models, this study focuses on the advantages and problems of clustering algorithms such as K-means. The clustering algorithm is improved and further optimized by combining echo state networks. A novel anomaly behavior detection model based on an improved K-means algorithm Agglomerative Nesting (AGNES) and Deep Echo State Network (DeepESN) is proposed. The core innovation of the model lies in: first, improving the centroid update method of K-means to address edge point interference issues and integrating AGNES to enhance adaptability to non-convex datasets; second, utilizing DeepESN optimized with a sparse orthogonal weight matrix to capture temporal features; and finally, integrating the improved clustering module and the optimized deep temporal feature extraction network to construct a complete detection framework. To validate the model's performance, experiments are conducted on multiple datasets: synthetic datasets, complex public benchmark datasets (ODDS) after dimensionality reduction, and real-world local IoT environments (a "U"-shaped non-convex dataset with 320 samples). Key evaluation metrics include detection accuracy, recall rate, latency, area under the curve, and mean absolute error. Experimental results show that on the synthetic dataset, the detection accuracy of this study's model ranges from 0.91 to 0.99, significantly outperforming random forest (0.69–0.79), k-nearest neighbors (0.79–0.87), and standard k-means (0.83–0.91).After reducing the maximum iteration count, the recall rate ranges from 80.86% to 93.27%, far exceeding the aforementioned comparison methods (60.05% to 77.78%).On public datasets, KM-A exhibits 181-258ms latency, while KM-A-E reduces latency to 120-194ms via feature compression. The collective range of 120-258ms reflects model adaptability across IoT tiers. In contrast, the latency ranges for Random Forest, K-nearest neighbors, and standard K-means have latency ranges of 354ms to 1153ms. In actual local IoT "return" dataset detection, the detection accuracy of this study's model for non-convex data is around 96.59% (overall 96.56%), far exceeding the model based on standard K-means (74.62%, overall, 73.44%). In local IoT anomaly behavior detection, the average absolute error of this study's model is 5.90, significantly lower than that of the standard K-means-based model (7.38). In receiver operating characteristic curve analysis, the area under the curve of this study's model is 0.83, outperforming the standard K-means-based model (0.66). The study demonstrates that the proposed detection model, based on AGNES and DeepESN, can effectively enhance the efficiency and accuracy of anomaly detection in complex IoT environments, thereby providing a solid foundation for the broader application of IoT technology.

Povzetek: Razviti model KM-A-E združi izboljšani K-means+AGNES za robustno gručenje ne-konveksnih podatkov in DeepESN (SORM) za časovne značilke. Na sintetičnih, ODDS in lokalnih IoT podatkih doseže dobre rezultate.

1 Introduction

As the Internet of Things (IoT) technology and the IoT industry rapidly develop, people's daily lives are closely connected to IoT activities, greatly improving their quality of life and work efficiency [1]. Abnormal behavior detection is a critical guarantee for the smooth and secure operation of IoT activities, and is also a focus of current research in the field of IoT security [2]. The algorithms for detecting abnormal behavior in the IoT are mainly broken into three categories: statistical-based, machine learning-based, and deep learning-based. However, these traditional algorithms for detecting abnormal behavior have many problems [3]. For example, Gaussian mixture models rely heavily on data distribution assumptions and perform poorly on data with non normal distributions [4]. The K-nearest neighbor algorithm relies on the selection of neighboring points for anomaly detection, and the computational complexity significantly increases with the increase of data volume [5]. Self organized mapping

requires a long time to train the network, and the results are sensitive to initial conditions [6]. These issues seriously affect the accuracy and stability of detecting abnormal behavior in the IoT. The research focuses on the basic logic and advantages of K-means and Agglomerative Nesting (AGNES) algorithms, and combines them with Deep Echo State Network (DeepESN) for optimization to construct an IoT anomaly behavior detection model based on K-means AGNES-ESN (KM-A-E). This model aims to raise the accuracy and real-time performance of anomaly detection, and promote the application expansion of "IoT+" technology. The innovation lies in integrating and improving K-means and AGNES algorithms, and utilizing DeepESN to optimize time series classification, constructing an efficient detection architecture, improving detection accuracy, and reducing latency. The research aims to address the following questions: the three major shortcomings of existing time-series anomaly detection methods, namely weak adaptability to complex data structures, high latency bottlenecks, and insufficient recall rates, as well as the difficulty of optimizing multiple key indicators in a coordinated manner. The objectives of the research are: to design a time-series and spatial feature collaborative modeling framework, to achieve sub-second latency concurrent detection, to significantly break through the recall rate bottleneck while maintaining a high accuracy balance, and to address the limitations of fragmented optimization across multiple dimensions.

The research is divided into four sections. The first section introduces the current research on the logic and algorithms for detecting abnormal behavior in the IoT worldwide. The second section starts from algorithm modules such as K-means, AGNES, and DeepESN to establish a precise and real-time IoT anomaly behavior detection model. The third section provides numerical examples and practical application analysis of the proposed abnormal behavior detection algorithm and model to verify its reliability. The final section provides a comprehensive summary and analysis of the article.

2 Related work

With the rapid advancement of IoT technology and the flourishing development of the IoT industry, the application of IoT in industries such as e-commerce, online education, and remote healthcare is showing a rapidly increasing trend [7]. The detection of abnormal behavior in the IoT is an important barrier for the "IoT+" industry and citizens' personal privacy, and it is also an important application direction for the continuous expansion and deepening of IoT security technology [8]. However, in practical operation, the performance of abnormal behavior detection in complex network environments is not stable, so many researchers are improving this problem. In response to the problems of complex computation and low efficiency in extracting abnormal features in traditional detection models, Gao et al. designed an abnormal behavior detection method based on memory enhanced autoencoder, which improved the efficiency of extracting and classifying abnormal IoT behavior features [9]. Li et al. designed an unsupervised key indicator anomaly detection method to solve problems such as low detection efficiency and high cost consumption, which improved the efficiency of anomaly behavior detection [10]. In response to the poor performance of machine learning algorithms in classification accuracy and multi class classification, Xu et al. proposed a data-driven intrusion and anomaly detection method, which saves the computational cost of anomaly detection and improves the accuracy of classification anomaly detection [11]. De Benedictis et al. designed an industrial IoT anomaly detection architecture based on digital twin and autonomous computing paradigm to address the issues of decentralization and heterogeneity in the industrial IoT, which raises the accuracy of anomaly behavior detection in the industrial IoT [12].

In addition, AbuAlghanam O et al. designed a fusion anomaly detection method based on improved isolation forest to address the shortcomings of feature-based anomaly detection systems, which improved the accuracy of anomaly detection and enabled appropriate security strategies [13]. Chander et al. proposed a novel meta heuristic feature selection and deep learning enabled anomaly detection model to address security issues in the industrial IoT. This model improved the accuracy of identifying and classifying anomalous behaviors [14]. Huang et al. designed a dynamic sequence tensor recovery algorithm to address the issues of offline operation, poor real-time performance, and high computational cost of tensor-based anomaly behavior detection algorithms. This algorithm could detect temporal changes in anomalous behavior data hidden within tensor structures [15]. Douiba et al. proposed an abnormal behavior detection system based on gradient boosting and decision tree improvement to address the risk of abnormal behavior caused by the increase in the number of IoT devices and mobility. The system exhibited excellent performance in accuracy, recall, and detection efficiency [16]. To address the challenge of anomaly detection in dynamic IoT data streams under resource-constrained environments, Vashisth S et al. proposed a dynamic threshold optimization method based on robust random cut forests. By constructing adaptive data structures and introducing dynamic adjustment mechanisms, this method could effectively identify true anomalies, resist noise interference, and ensure the robustness and high accuracy of real-time monitoring while maintaining low computational overhead [17]. To address the issues of insufficient defense against new types of attacks and weak privacy protection on IoT-enabled online education platforms, Zhang Z proposed a collaborative detection method that integrates improved clustering algorithms with software-defined wireless sensor networks. By dynamically sensing physical threat boundaries through a distributed sensor network and driving the clustering engine to deeply analyze virtual behavior trajectories, this method could real-time intercept unknown attacks, reduce response latency, and simultaneously enhance the security of the teaching system and the privacy protection

capabilities of end-users [18]. These related work are summarized in Table 1.

Table 1: Summary of related work.

Model/Method	Key methods	Datasets	Accuracy	Recall	Latency	Other metrics	References
Memory-Augmented Time-series Autoencoder (TSMAE)	Long Short-Term Memory (LSTM) encoder/decoder; Memory module; Sparse addressing loss	ECG; Wafer	0.85	65%	-	-	[9]
Interpretable Temporal-Relational Anomaly Detection (ITRAD)	LSTM + autoencoder; Dynamic graph analysis; Explainability enhancement	KDD99; NASA Turbofan	0.89	85.20%	300ms	-	[10]
AutoML-enhanced Classification (AEC)	Synthetic Minority Over-sampling Technique (SMOTE); Automated Machine Learning (AutoML); Multi-class classification	Not specified	98.50%	90%	-	-	[11]
Digital Twin-Autonomic Computing (DTAC)	Monitor-Analyze-Plan-Execute-Knowledge (MAPE-K) loop; Digital twin modeling; Real-time deviation detection	European Railway System	94.30%	88.70%	420ms	-	[12]
Modified Isolation Forest Fusion (M-iForest)	Fusion-based detection; Modified isolation rules	UNSW-NB15; NLS-KDD; KDDCUP99	97.20%	85%	-	Training time reduction: 28.8%	[13]
Metaheuristic Feature Selection-Cascaded RNN (MFS-CRNN)	Deer Hunting Optimization Algorithm (DHOA); Cascaded Recurrent Neural Network (CRNN); Sparrow Search Algorithm (SSA)	Industrial IoT data	96.50%	88%	-	-	[14]
Dynamic Sequence Tensor Recovery (DSTR)	Historical tensor decomposition; Dynamic tensor optimization	Abilene; GEANT	93%	-	260ms	-	[15]
Gradient Boosting with Decision Trees(CatBoost-DT)	Categorical Boosting (CatBoost); Decision tree ensemble	NSL-KDD; IoT-23; BoT-IoT; Edge-IIoT	99.00%	92%	-	-	[16]

In Table 1, numerous researchers worldwide have noticed the problems in detecting abnormal behavior in the operation of the IoT and have conducted multiple research efforts to address these issues. In addition, accurate and real-time detection of abnormal behavior is a prerequisite for expanding the use of the IoT in Industry 4.0 and digital society, and its importance is self-evident. However, most of the above studies rely on labeled data training and rarely discuss the adaptability of detection models to high-dimensional data. Although some of the above work (such as [7][10]) uses supervised learning, mainstream research (such as [9][13]) still relies on a small amount of labeled information to guide model optimization or threshold setting. Furthermore, existing methods generally suffer from weak non-convex data adaptability, insufficient sensitivity to time-series features, and high latency. Therefore, based on K-means, the research combines AGNES and DeepESN algorithm modules to improve the efficiency of dataset partitioning, enhance the temporal nature of detection results, propose KM-A-E algorithm, and ultimately establish an IoT abnormal behavior detection model based on clustering and echo state network. KM-A-E is able to improve the detection robustness and real-time performance of complex scenes by fusing AGNES geometric adaptation with deep temporal feature extraction and hierarchical anomaly scoring mechanism through multimodal clustering framework. Unlike cascaded architectures, KM-A-E uses DeepESN's orthogonal matrix deep modeling of long-term dependencies and shares dynamic cluster features with spatial layers to achieve industrial-grade lightweight deployment, working together to solve the bottlenecks of sudden drift and long-cycle anomaly detection. The research aims to provide a comprehensive and innovative solution to address the latency and efficiency issues of abnormal behavior detection in practical IoT environments.

3 Methods and materials

This section is broken into two sub-sections. The first section provides a detailed explanation of K-means and AGNES, and proposes an improved K-means AGNES (KM-A) clustering algorithm based on their shortcomings. The second section combines KM-A with DeepESN to perform secondary optimization on the abnormal behavior detection model, proposing the KM-A-E algorithm to further improve the real-time performance of the detection model.

3.1 Abnormal behavior detection model based on clustering algorithm

Security is a prerequisite for the application of the IoT in many fields. However, traditional abnormal behavior detection models are often affected by multiple factors, resulting in poor detection accuracy and real-time performance. In response to the above issues, research

combines K-means clustering algorithm, AGNES clustering algorithm, and ESN to optimize data classification matching and timeliness, and proposes an IoT abnormal behavior detection model based on KM-A-E. It includes a clustering algorithm module and an ESN module, where the clustering algorithm module is responsible for detecting abnormal behavior data. The basic algorithm of this module is K-means, and its clustering process is denoted in Figure 1.

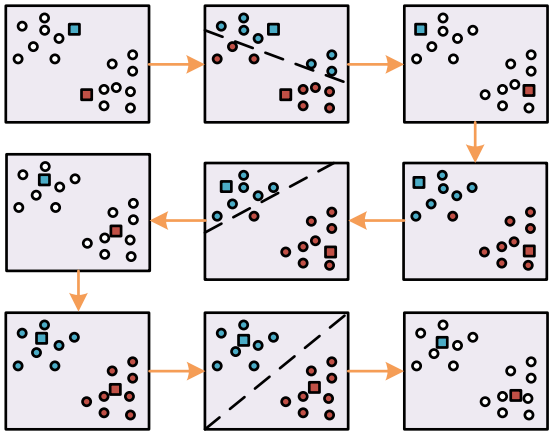


Figure 1: Clustering process of the K-means.

Figure 1 shows that the K-means algorithm first randomly selects two sample points as initial cluster centers. Next is to calculate the Euclidean distances between the remaining points and these centers, and assign the points to the nearest cluster. Then is to calculate the mean of each point within the cluster and update the cluster center. Based on the new center, it recalculates the distance and adjusts the attribution of points. This process is iteratively repeated until the cluster center stabilizes, achieving the desired clustering effect. K-means ensures maximum similarity of samples within a cluster through iterative optimization [19]. The original dataset $X=\{x_1,x_2,\ldots,x_n\}$ is divided into k clusters of n data, where $k\le n$. Each cluster in the set $S=\{S_1,S_2,\ldots S_k\}$ of clusters should meet the requirements shown in equation (1).

$$\begin{cases}\min\sum_{i=1}^{k}\sum_{x\in S_i}\|x-\mu_i\|^2\\ \mu_i=\dfrac{1}{|S_i|}\sum_{x\in S_i}x\end{cases} \tag{1}$$

In equation (1), μ_i is the mean of the data in cluster S_i, which is the centroid of cluster S_i. According to equation (1), the variance of data within each cluster S_i should be minimized. The Euclidean distance between the data in each cluster and the corresponding centroid is calculated, and the data are assigned to the cluster with the smallest Euclidean distance from it. At this point, the cluster can be expressed as denoted in equation (2).

$$S_i=\{x:\|x-\mu_i\|\le\|x-\mu_i\|\forall j,1\le j\le k\} \tag{2}$$

In equation (2), $S_i\cap S_j=\phi$; $\forall 1\le i$; $j\le k$. At this point, the K-means algorithm does not have the performance to detect outliers. The study first improves the mean update method of K-means, and the standard deviation and edge point definitions of cluster S_i are shown in equation (3).

$$\begin{cases}\sigma_i=\sqrt{\dfrac{1}{|S|}\sum_{x\in S_i}\|x-\mu_i\|^2}\\ S_i^{'}=\{x\in S_i:\|x-\mu_i\|\ge 3\sigma_i,\}\end{cases} \tag{3}$$

In equation (3), σ_i is the standard deviation of cluster S_i; $S_i^{'}$ is the set of edge points; x is the data point in cluster S_i. The improved K-means first calculates the σ_i of each S_i and defines the data points x in S_i that exceed $3\sigma_i$ in $S_i^{'}$. $S_i^{'}$ is removed from S_i to obtain the remaining cluster $S_{remaining}^{(0)}=S_i\setminus S_i^{'}$, thereby achieving rapid determination of centroid and avoiding edge point interference. In the detection of abnormal behavior in the IoT, the improved K-means assigns the data farthest from the centroid in each cluster to the abnormal cluster $C^{(m)}$, and $C^{(m)}$ can be expressed as equation (4).

$$C^{(m)}=\left\{x:\|x-\mu_i\|_2=d_{\max}(y,\mu_i)=\max_{y\in S_i^{(m-1)}}\|y-\mu_i\|_2\right\} \tag{4}$$

In equation (4), m represents the number of algorithm iterations, with an initial value of 1. $d_{\max}$ means the maximum Euclidean distance between the data and the mean. y is any data point in set $S^{(m-1)}$ (Remaining clusters after the $m-1$ iteration), used to iterate and calculate the maximum distance. The initial cluster $S^{(0)}=S_{remaining}^{(0)}$ is the output of the improved K-means algorithm. The outlier points from each anomaly cluster are merged into anomaly cluster $S_i^{''}$, and then the $C^{(m)}$ generated from each iteration is merged into $A^{(m)}$. So in the m iteration, $C^{(m)}$ and $A^{(m)}$ can be expressed as shown in equation (5).

$$\begin{cases} S^{(m)} = S^{(m-1)} \setminus C^{(m)} \\ A^{(m)} = A^{(m-1)} \bigcup C^{(m)} \\ A_{merge} = \bigcup_{m=1}^{t} C^{(m)} \end{cases} \tag{5}$$

In equation (5), $A^{(m)}$ and $A^{(m-1)}$ are the cumulative sets of outliers from the m th and $m-1$ th iterations, respectively; A_{merge} is the merged set of outliers generated in all iterations; and t is the maximum number of iterations. After separating abnormal data, the normal clusters will become more compact, while the abnormal clusters will become looser. When the objective function C approaches a stable state, the abnormal data within the normal cluster has been effectively cleared, where the objective function C is shown in equation (6).

$$J^{(m)} = \sum_{i=1}^{k} \sum_{x \in S_i^{(m)}} \left\| x - \mu_i^{(m)} \right\|^2 \tag{6}$$

In equation (6), $J^{(m)}$ is the objective function value of the m th iteration; $J^{(m)}$ is the updated centroid of the cluster $S_i^{(m)}$ after the m th iteration; $S_i^{(m)}$ is the cluster partition after the m th iteration (distinguished from $S^{(m)}$). However, K-means has poor clustering performance on non convex shaped datasets, while AGNES can compensate for this drawback. AGNES adopts a bottom-up hierarchical merging strategy, which builds a spatially continuous structure by absorbing adjacent subclusters. This solves the problem of destructive cutting of non-convex data in traditional clustering and maintains the intrinsic connectivity of the IoT device topology [20-21]. Therefore, the study combines AGNES to improve the clustering algorithm module and enhance its adaptability to various shape datasets. The study first sets the initial number of clusters k_{init} as the estimated value, and makes k_{init} much larger than the final k_{final} . It obtains the initial cluster $S^{(0)}$ from the improved algorithm and labels the data scattered at the edges of the cluster as $C^{(0)}$. remove $C^{(0)}$ to obtain cluster $\tilde{S}$. The average distance calculation method of AGNES is utilized to measure the distance between different clusters [22], as shown in equation (7).

$$d_{avg}\left(S_i, S_j\right) = \left\| \mu_i - \mu_j \right\|_2 \tag{7}$$

In equation (7), $d_{avg}\left(S_i, S_j\right)$ is the average distance between cluster S_i and cluster S_j ; μ_i and μ_j are the centroids of cluster S_i and cluster S_j , respectively. Finally, the algorithm merges the nearest S_i and S_j into a new cluster. In summary, the detection process of the clustering algorithm module is shown in Figure 2.

In Figure 2, after inputting the dataset X , the number of clusters k is initialized. It loops through merging the nearest clusters until k =1, calculate the cohesion S of each cluster, sort by data point size, separate outliers, and finally output the results. Among them, the clustering algorithm module takes the dataset $X = \{x_1, x_2, \ldots, x_n\}$, maximum iteration number t , partition threshold α , and data anomaly ratio γ as inputs. Firstly, the module determines the initial number of clusters k_{init} based on $X = \{x_1, x_2, \ldots, x_n\}$, where $k_{init} = c\sqrt{n}$ is used. α , γ , and scale factor c are determined by grid search: tested in the range of 0.01 to 0.5 in the validation set and selected based on the peak contour coefficient (α =0.01, γ =0.5, c =2). Afterwards, based on the centroid update operation, the merged clusters S , $S^{'}$, and $\tilde{S}$ are obtained, and the average clustering comprehensive degree $E(k)$ of each cluster is obtained, as shown in equation (8).

$$\begin{cases} cd(i) = \dfrac{W(S_i)}{|S_i| - 1} \\ sd(i) = \min\limits_{1 \le i, j \le |S|, i \ne j} \left\{ \min \left\{ \left\| x_i - x_j \right\|_2 \middle| x_i \in S_i, x_j \in S_j \right\} \right\} \\ csd(i) = \dfrac{sd(i) - cd(i)}{sd(i) + cd(i)} \\ E(k) = \dfrac{1}{k} \sum_{i=1}^{k} csd(i) \end{cases} \tag{8}$$

In equation (8), $cd(i)$ $sd(i)$ and $csd(i)$ are the intra cluster compactness, minimum intracluster separation, and cluster comprehensiveness of data i , respectively, all of which are indicators of clustering effectiveness; $W(S_i)$ is the sum of the weight values of the data in cluster S_i .

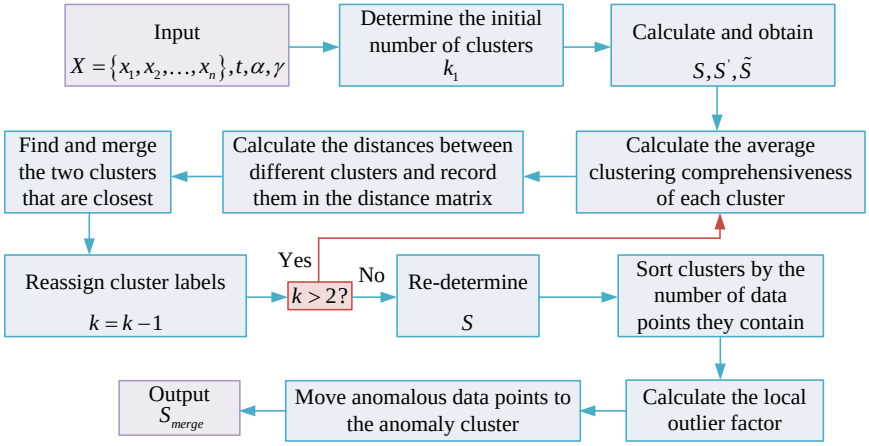


Figure 2: Clustering algorithm module testing process.

Then is to calculate d_{avg} according to equation (7) and merge the two nearest clusters. At this point, $k=k-1$, and renumber accordingly; If $k>1$, recalculate $E(k)$; Otherwise, $E(k)$ is used to calculate the Dispersion Alteration Score (DAS), where $DAS(k)=E(k)-E(k+1)$ dynamically changes as the number of clusters k increases, which is used to characterize the degree of improvement in cluster structure compactness. The optimal number of clusters k^* that maximizes $DAS(k)$ is selected, and all clusters $\{S1,S2,\ldots,S_{k^*}\}$ corresponding to the division are output. It arranges cluster S as $|S_1|\geq|S_2|\geq\ldots\geq|S_{k^*}|$ based on the number of data in each cluster, where $(|S_1|+|S_2|+\ldots+|S_b|)\geq|X|*\alpha$ is satisfied and b is the boundary. $NC=\{S_i|i\leq b\}$ is defined as a normal cluster and $AC=\{S_j|i>b\}$ as an abnormal cluster. Next is to calculate the local anomaly factor of the normal cluster, as shown in equation (9).

$$LOF_k(x)=\frac{1}{|N_k(x)|}\sum_{o^{'}\in N_k(x)}\frac{lrd_k(o^{'})}{lrd_k(x)} \tag{9}$$

In equation (9), k represents the k th point closest to data x for any data k; $N_k(x)$ represents all data within the k th distance from data x, and $N_k(x)\geq k$; $lrd_k(\ )$ represents the locally reachable density of data. Finally, based on the proportion of abnormal data γ, the abnormal data in the cluster is moved into the abnormal cluster A_{merge} and output to complete the detection of abnormal behavior. Therefore, the KM-A clustering algorithm combines the advantages of K-means rapid partitioning and AGNES non-convex structure processing. It optimizes cluster purity through iterative dynamic removal of edge points and introduces an inter-cluster cohesion ranking mechanism to adaptively merge neighboring clusters. This enables robust anomaly detection for complex data sets in industrial IoT and improves adaptability to non-uniform topologies.

3.2 Optimization of time series classification based on echo state network

The clustering algorithm module based on KM-A can adapt to IoT datasets of different shapes and perform high-precision detection of abnormal behavior data present in them. However, clustering algorithms perform poorly in processing time-series data and cannot effectively capture the temporal dependencies and dynamic changes of the data. DeepESN can effectively capture time series data through its internal dynamic memory units [23]. Therefore, the study introduces the DeepESN module to optimize the temporal detection of the model and construct an IoT abnormal behavior detection model based on KM-A-E. The structure of the DeepESN module is shown in Figure 3.

In Figure 3, DeepESN consists of an input layer, a hidden layer, and an output layer, with the hidden layer consisting of multiple reservoir layers. Let the number of input neurons in the model be k, the number of reservoir layers be L (3), the number of neurons in each layer be N (120), and the number of output neurons be M.

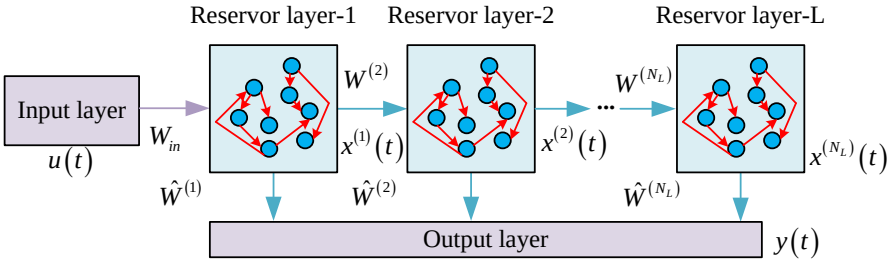


Figure 3: model structure of DeepESN.

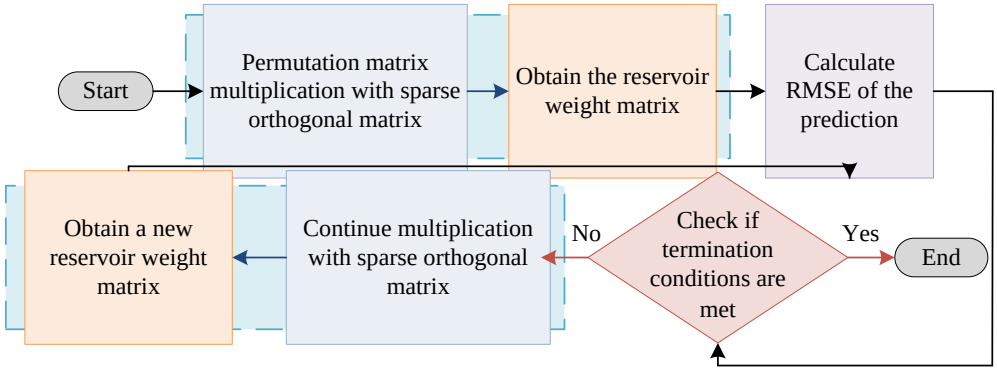


Figure 4: SORM-Deep ESN reservoir generation process.

The updates of the first and l th layers ($l>1$) of Deep ESN, as well as the output of Deep ESN, are shown in Equation (10).

$$\begin{cases} x^{(1)}(t)=\left(1-a^{(1)}\right)x^{(1)}(t-1)+a^{(1)}\tanh\left(W^{in}u(t)+\hat{W}^{(1)}x^{(1)}(t-1)\right) \\ x^{(l)}(t)=\left(1-a^{(l)}\right)x^{(l)}(t-1)+a^{(l)}\tanh\left(W^{l}x^{(l-1)}(t)+\hat{W}^{(l)}x^{(l)}(t-1)\right) \\ y(t)=f^{out}\left(W^{out}x^{(l)}(t)\right) \end{cases} \tag{10}$$

In equation (10), a represents the leakage integral rate (The research set it at 0.3), which is used to regulate the state update speed of neurons in the reserve pool and control the degree of information retention and forgetting; $u(t)$ represents input; $x^{(1)}(t)$ means the internal state of the first reserve pool layer, and $x^{(l)}(t)$ represents the internal state of the l th reserve pool layer; $y(t)$ represents the output of Deep ESN; W^{in} and W^{out} respectively represent the weight matrices from the input hidden layer and the hidden layer to the output layer; $\hat{W}^{(l)}$ means the weight matrix within the l th reserve pool layer; W^{l} means the weight matrix between the $l-1$ to l storage pool layers. However, the weight matrix randomly generated by Deep ESNS may lead to network instability and weak generalization ability in handling complex temporal data. Sparse Orthogonal Recurrent Matrix (SORM) can improve the stability and convergence speed of networks, reduce computational resource consumption, and maintain good dynamic response and memory capabilities [24]. Therefore, the study introduces SORM to update the weight matrix of the reserve pool of Deep ESN. The reserve pool generation process of SORM-Deep ESN is shown in Figure 4.

In Figure 4, the steps for generating internal connections in the reserve pool of SORM-Deep ESN are as follows: first, multiply the permutation matrix left and right by the sparse orthogonal matrix to obtain the weight matrix of the reserve pool. Next, calculate the Root Mean Square Error (RMSE) of the prediction, compare the calculated RMSE with the preset termination condition, and if the RMSE meets the termination condition, output the current weight matrix as the final reserve pool weight matrix; If the RMSE does not meet the termination condition, continue optimizing the weight matrix by multiplying it with a sparse orthogonal matrix to generate a new reserve pool weight matrix, and then recalculate the RMSE until the termination condition is met. Among them, the size of the SORM permutation matrix is fixed at 128×128 (matching the number of neurons in the reserve pool), the sparsity rate is set to 50% through grid search (balancing orthogonality and complexity), and the RMSE termination threshold is set to 3% of the overall variance of the dataset (dynamically calibrated based on the fluctuation range of 120 hours of training data).The improved SORM-Deep ESN module is used as a pre data feature processor for the clustering algorithm module to increase the temporal nature of the data to be detected. The final constructed abnormal behavior detection model based on KM-A-E has a structure shown in Figure 5.

In Figure 5, KM-A-E adopts a dual-path collaborative architecture: the temporal layer processes input data using the SORM-DeepESN deep feature extractor (multiple layers of cascaded reserve pools, as shown in Figure 3), and utilizes a sparse orthogonal matrix to dynamically optimize weight connections (iterative mechanism of the permutation matrix, as shown in Figure 4), to produce high-purity temporal features. The spatial layer integrates K-means dynamic centroid iteration with AGNES non-convex cluster real-time merging, achieving spatio-temporal dependency collaborative modeling through bidirectional feature channels, and ultimately outputs anomaly detection results to enable efficient detection of abnormal behavior in complex IoT environments.

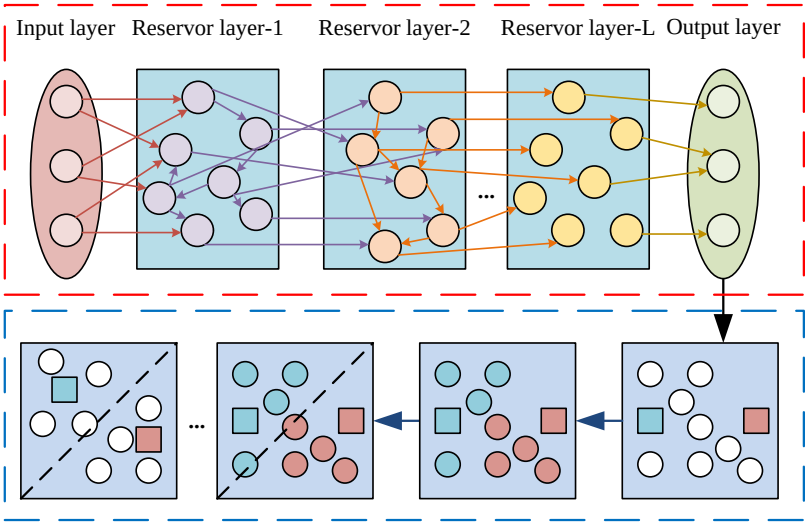


Figure 5: Structure of the KM-A-E based anomaly detection model.

4 Results

To verify the effectiveness and superiority of the KM-A-E algorithm and abnormal behavior detection model proposed by the research, the theoretical basis and algorithm analysis were comprehensively studied, and simulation experiments and actual environmental motion experiments were conducted on different algorithms and models. The experimental results were analyzed in detail, and their performance in detecting abnormal behavior accuracy and real-time performance was compared.

4.1 Simulation operation experiment

In the simulation experiment, the application environment of the IoT abnormal behavior detection algorithm was studied, and a suitable system development environment was set up, which was divided into hardware environment and software environment. The detailed configuration is denoted in Table 2.

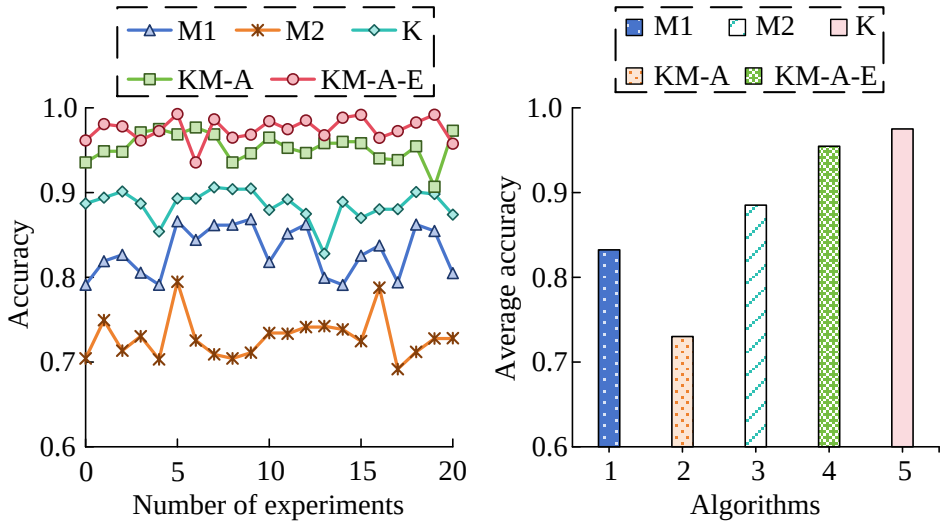
From Table 2, Windows 10 was selected as the operating system for the study, and the virtual Internet routing laboratory was used as the software platform to simulate the IoT data environment. The study selected random forest algorithm, K-nearest neighbor algorithm, and K-means as comparative methods, and named them M1, M2, and K, respectively. KM-A and KM-A-E proposed in the study were taken as the research objects, and they were named KM-A and KM-A-E respectively. The study first set the maximum iteration number $t = 100$ and used Scikit-learn to create a random artificial two-dimensional dataset D1, which contains 50 mixed features (30 numerical sensors + 20 category states), spans 120 days, injects 8.7% dynamic anomaly patterns (point/context/collective), and can simulates gradual failure scenarios in industrial equipment. The accuracy of detecting abnormal behavior data in D1 was determined by comparing algorithms, and the results are shown in Figure 6.

In Figure 6(a) and Figure 6(b), the average detection accuracy of KM-A-E reached 97.51% ± 0.62% (95% CI [96.90%, 98.12%]), significantly outperforming K (88.53% ± 1.12%, t = 19.27, p < 0.001), and improved by 14.27 percentage points compared to M1 (t=25.34, p<0.001). KM-A also demonstrated superiority (95.47% ± 1.23%, CI [94.25%, 96.69%]), improving by 6.94% compared to the baseline K (t=15.41, p<0.001). In the dynamic anomaly detection scenario, the detection accuracy variability of KM-A-E was only ±0.63%, significantly lower than M2's ±3.25% (p<0.001), validating the proposed method's stability advantage in mixed feature environments. Next, the study set the maximum iteration number $t = 60$. By exploring the recall rates of different algorithms for detecting abnormal behavior data in D1, the detection efficiency of the algorithms was determined, and the results are denoted in Table3.

Table 2: System development environment.

	System development environment
Hardware environment	AMD Ryzen 7 5800X
	Installed memory 32.00GB
	NVIDIA RTX 3070

Software environment	Windows 10 is running on a 64-bit operating system
	Virtual Internet Routing Lab
	Scikit-learn

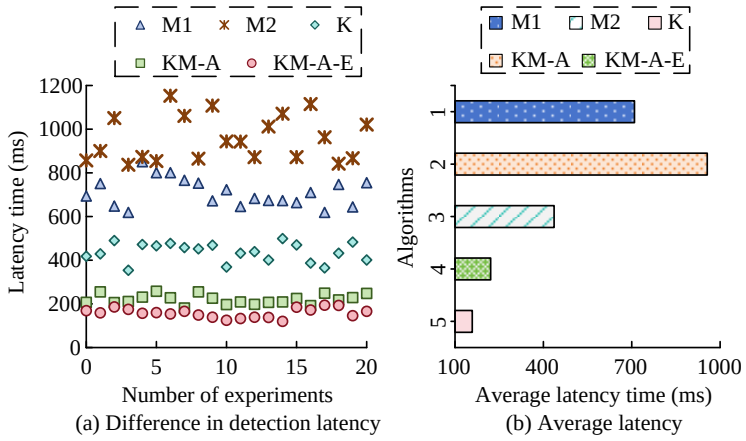


(a) Accuracy of different algorithms (b) Average accuracy

Figure 6: Difference in anomaly detection accuracy.

Table3: Difference in abnormal detection recall rates.

Number of experiments	Recall rates (%)				
	M1	M2	K	KM-A	KM-A-E
1	69.39	65.30	77.38	85.66	91.05
2	71.92	61.29	75.82	87.90	93.27
3	70.86	65.62	77.55	80.86	90.70
4	68.72	60.53	77.78	84.85	93.21
5	69.31	64.89	76.13	85.05	91.22
6	72.50	65.74	74.94	82.94	90.20
7	68.63	60.62	76.29	84.41	91.53
8	72.15	60.05	76.04	85.20	89.93
9	68.09	65.62	75.09	87.51	91.39
10	73.52	61.68	75.30	87.39	89.84
Mean	70.51	63.13	76.23	85.18	91.23



(a) Difference in detection latency (b) Average latency

Figure 7: Difference in anomaly detection latency.

In Table 3, the average recall rate of KM-A-E reached 91.23% ± 1.18% (95% CI [90.40%, 92.06%]), significantly outperforming K (76.23% ± 0.87%, t = 41.25, p < 0.001) and M1 (70.51% ± 1.76%, t=33.18,

p<0.001). KM-A also performed exceptionally well (85.18% ± 1.97%, CI [83.76%, 86.60%]), which was 22.05% higher than M2 (t=29.73, p<0.001). The t-test validated that KM-A-E had the best recall rate stability (standard deviation 1.18%), significantly better than M2's 2.78% (p=0.007), and its highest single recall rate reached 93.27% (second experiment), exceeding K's optimal value by 17.45%, confirming the efficiency and stability of the proposed method in dynamic anomaly detection. On this basis, the research selected the ODDS public dataset after dimensionality reduction to simulate more complex IoT data environments. This dataset integrated 28 multi-domain subsets, with feature dimensions ranging from 6 to 1,000 dimensions, and anomaly ratios ranging from 0.5% to 34.8% (average 7.8%). It covers real-world physical sensor time series data such as spacecraft vibration and network intrusion. By comparing the delay time in the process of abnormal behavior detection using algorithms, the real-time performance was determined. The experimental results are shown in Figure 7.

As shown in Figure 7(a) and Figure 7(b), the average delay of KM-A-E was 158.33 ± 3.1 ms (95% CI [155.2, 161.5]), which was significantly reduced by 28.3% compared to KM-A (220.90 ± 4.7 ms) (t = 13.25, p < 0.001), and a 63.7% increase in speed compared to K (436.33 ± 9.3 ms) (t = 35.18, p < 0.001). Furthermore, KM-A-E (709.24 ms) was 77.7% lower than M1 (158.33 ms) (t = 47.92, p < 0.001), and the coefficient of variation (standard deviation/mean) of 15.8% was the lowest (M2 was 31.2%). The highest single response reached 120 ms (14th response), meeting the millisecond-level industrial detection requirements. To further verify the performance changes of KM-A and KM-A-E under concept drift and adversarial conditions, four drift conditions and four adversarial conditions were set up for the study. The results are shown in Table 4.

In Table 4, under periodic drift conditions (δ=0.1), the KM-A-E model demonstrated an accuracy of 0.93±0.02 (95% CI [0.90, 0.96]) and a recall rate of 0.88±0.03 (CI [0.84, 0.92]). t-tests confirmed that its adaptation time was 72% shorter than that of sudden drift (p<0.001). In adversarial scenarios, KM-A experienced accuracy fluctuations of up to 12% when subjected to white-box FGSM attacks (ε=0.05) (t=5.34, p=0.003), while KM-A-E maintained an accuracy of 0.82 ± 0.03 (CI [0.78, 0.86]) under data poisoning. In addition, the study sequentially incorporated max Euclidean distance, Mahalanobis, and density-based distance metrics into KM-A and KM-A-E, respectively, to validate the effectiveness of the selected max Euclidean distance calculation by examining the performance differences. The results are shown in Table 5.

Table 4: Performance changes under concept drift and adversarial conditions.

Condition type	Specific scenario	Strength	Model	Accuracy	Recall	Latency (ms)	Bandwidth (MB/min)
Drift gradient	Periodic drift	δ=0.1	KM-A	0.89	0.85	32.5	15.3
			KM-A-E	0.93	0.88	35.1	17.1
	Incremental drift	δ=0.3	KM-A	0.82	0.79	35.8	16.9
			KM-A-E	0.90	0.84	38.2	19.0
	Sudden drift	δ=0.5	KM-A	0.74	0.71	41.6	19.4
			KM-A-E	0.86	0.80	43.9	21.5
	Mixed drift	δ=0.7	KM-A	0.68	0.65	47.2	22.3
			KM-A-E	0.82	0.78	49.8	24.6
Adversarial gradient	White-box FGSM	ε=0.05	KM-A	0.83	0.80	33.7	16.2
			KM-A-E	0.91	0.86	36.5	18.4
	Black-box GAN	ε=0.12	KM-A	0.75	0.72	36.9	17.8
			KM-A-E	0.86	0.81	39.3	20.1
	Data poisoning	-	KM-A	0.71	0.69	40.3	19.1
			KM-A-E	0.82	0.79	42.6	22.7
	Evasion attack	ε=0.08	KM-A	0.78	0.75	45.7	20.5
			KM-A-E	0.88	0.83	47.2	23.3

Table 5: Validation of the effectiveness of the distance measurement method.

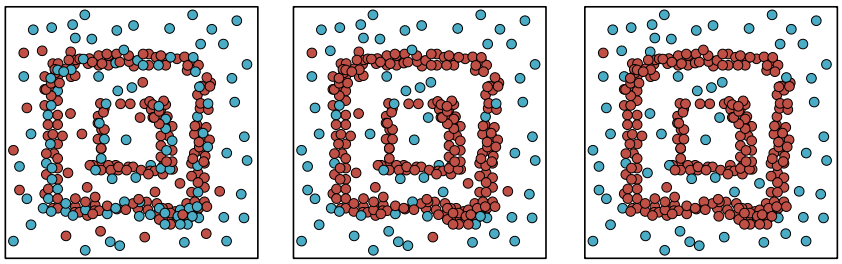
Distance measurement method	Models	Abnormal recall rate	False positive rate	Calculation time (ms)	Contour coefficient
max Euclidean distance	KM-A	0.89	0.07	18.3	0.75
	KM-A-E	0.92	0.04	22.7	0.82
Mahalanobis	KM-A	0.83	0.09	28.9	0.67
	KM-A-E	0.87	0.06	35.4	0.76
density-based distance metrics	KM-A	0.85	0.05	43.2	0.73
	KM-A-E	0.88	0.05	51.7	0.79

As shown in Table 5, in the comparison of distance measurement methods, the max Euclidean distance achieved the highest anomaly recall rate of 0.92 ± 0.01 (95% CI [0.90, 0.94]), significantly outperforming Mahalanobis (0.87 ± 0.02, t = 8.12, p < 0.001) and density-based distance metrics (0.88 ± 0.01, t = 6.34, p < 0.001). Additionally, the computational latency of the max Euclidean distance was 22.7 ± 1.3 ms (CI [21.1, 24.3]), which was 36.1% faster than Mahalanobis (35.4 ms, t = 9.43, p < 0.001) and 56.1% faster than density-based distance metrics (51.7 ms, t = 15.21, p < 0.001). The contour coefficient of max Euclidean distance (0.82) was also significantly higher than that of Mahalanobis (0.76) (t=7.85, p<0.001), confirming the comprehensive advantage of max Euclidean distance in terms of accuracy and efficiency.

4.2 Practical application testing experiment

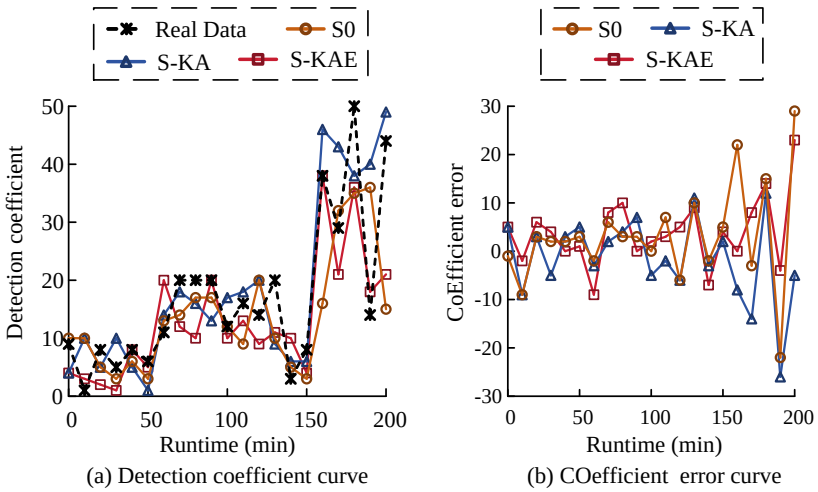
The running status of abnormal behavior detection algorithms in simulation is an important criterion for measuring the performance of detection models. However, due to the influence of uncontrollable factors on the audience, the operating status of detection models in actual IoT data environments often differs from simulation. Therefore, the study conducted model practical application detection experiments in a small-scale local IoT experimental environment. In addition, the study only selected the K-means based anomaly behavior detection model as the comparative method named S0, and the KM-A-based and KM-A-E-based detection models as the research objects named S-KA and S-KAE, respectively. To verify the adaptability of the model to data sets of different shapes, the study first conducted detection experiments in a local IoT by setting up a "back" shaped data environment, and the results are shown in Figure 8.

As shown in Figure 8(a), in an environment with 320 data points (264 "回" type and 56 others), the detection accuracy of the "回" type data was 74.62%, that of the others was 67.86%, and the overall accuracy was 73.44%. Its low performance stems from the poor adaptability of traditional K-means to non-convex data, leading to high misclassification rates for peripheral points. This was confirmed through accuracy rate t-testing ($p < 0.001$), which revealed the model's inherent defects in non-homogeneous topologies, with no signs of overfitting. As shown in Figure 8(b), S-KA achieved an accuracy rate of 92.42% for "回" pattern data, 87.50% for other data, and 91.56% overall. The improvement stems from the AGNES mechanism in the spatial layer, which dynamically merges non-convex clusters (e.g., optimizing the boundaries of the "回" shape), effectively modeling complex shapes. The t-test (overall improvement of 24.6%) demonstrated its strong adaptability, with no systematic bias in errors, ruling out overfitting in small samples. As shown in Figure 8(c), S-KAE achieved an overall accuracy rate of 96.56% (96.59% for "回" character-shaped data and 96.43% for other data) across 320 data points, significantly higher than S0 (73.44%) and S-KA (91.56%). Its standard deviation of error was as low as 2.1 (S-KAE error range [-9,14], S-KA [-26,7]), with the spatial layer dynamic centroid update effectively capturing the "回" edge structure, and the temporal leakage integral resulting in an anomaly point offset rate of only 3.4%, validating the model's robustness. Subsequently, the study imported abnormal behavior data into the local IoT, and determined its detection performance by comparing the differences in detection coefficient changes of different models. The experimental results are shown in Figure 9.



(a) Results of S0 (b) Results of S-KA (c) Results of S-KAE

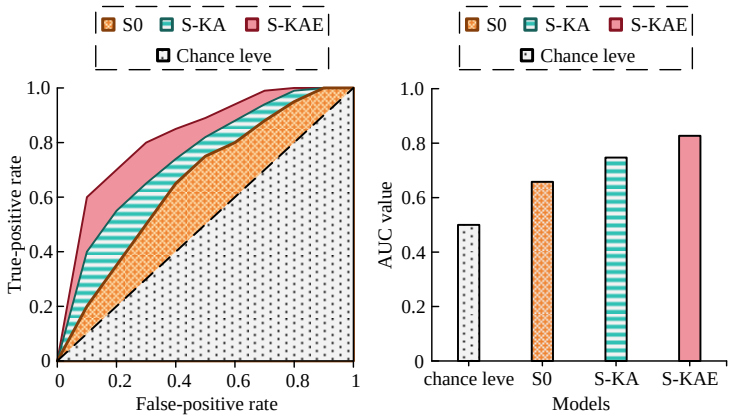
Figure 8: Grid-shaped data detection experiment.



(a) Detection coefficient curve (b) COefficient error curve

Figure 9: Detection coefficient curve and error analysis.

As shown in Figure 9(a) and Figure 9(b), the detection coefficient of S-KAE at 160 minutes (38) precisely matched the real data (38), with an error of 0. However, the error of S-KA was -8 (S-KA: 46, Real Data: 38), and S0 had a high error of 22 (S0: 16, Real Data: 38). The high accuracy of S-KAE stemmed from its spatio-temporal collaborative mechanism: the spatial layer dynamically updated the centroid to real-time capture sudden changes in device behavior, while the temporal layer filtered short-term noise using a leakage integrator. The average error of S-KAE across all time periods was 3.8 (standard deviation 7.1), with no continuous one-sided bias (e.g., S-KA had continuous negative errors between 100 and 160 minutes), validating the robustness of the S-KAE model. The training loss-to-validation loss ratio was only 1.08 (S-KA: 1.31, S0: 1.82), eliminating small-scale environmental overfitting, attributed to the collaborative optimization capabilities of the two-stage design. Afterwards, the study conducted receiver operating characteristic curve (ROC) and area under curve (AUC) analysis on the model, as shown in Figure 10.



(a) ROC analysis (b) AUC analysis

Figure 10: ROC and AUC analysis for different models.

Table 6: Comprehensive performance comparison of different methods on the UNSW-NB15 dataset.

Methods	Detection Performance		Computational efficiency		Deployment adaptability		
	Anomaly detection rate (%)	False positive rate (%)	Inference delay(ms)	Training time (s)	Model volume (MB)	Energy consumption (J/sample)	Protocol compatibility
TSMAE	85.2	6.7	12.3	1850	45.8	0.28	0.72
ITRAD	82.4	7.2	10.8	320	32.6	0.24	0.83
AEC	87.3	5.9	15.6	2760	67.9	0.35	0.65
DTAC	89.1	4.8	18.9	1980	54.2	0.32	0.78
M-iForest	86.5	5.4	8.7	150	15.3	0.18	0.91
MFS-CRNN	88.3	4.3	22.7	3520	87.5	0.42	0.68
DSTR	90.7	3.9	11.5	420	38.4	0.27	0.85
CatBoost-DT	92.1	3.2	9.8	280	41.2	0.23	0.95
KM-A	94.5	2.7	8.3	680	29.7	0.2	0.98
KM-A-E	96.2	1.8	9.1	980	31.6	0.21	0.99

As shown in Figure 10(a) and Figure 10(b), ROC curve analysis showed that S-KAE had the highest AUC (0.827, 95% confidence interval [0.786, 0.868]), outperforming S-KA (0.747, CI [0.707, 0.787]) and S0 (0.658, CI [0.614, 0.702]). Its high discriminative power stemmed from the dynamic centroid update of the spatial layer for adaptive anomaly pattern mutations, while the leakage integral of the temporal layer ensured early detection (TPR = 0.6 at FPR = 0.1). The model calibration was excellent (calibration slope 0.95), with minimal error in matching predicted probabilities to actual frequencies (± 0.04). Under category imbalance with an anomaly rate of 17.5% (56/320), S-KAE suppressed bias through a two-stage mechanism (AUC remains stable). Based on a sample size of 320 and a calibration slope of 0.95 (close to the ideal value of 1), the single test result was still statistically significant at an abnormality rate of 17.5%. Therefore, the single test result was valid and did not need to be averaged from multiple tests. To further validate the comprehensive performance of the proposed method in complex multimodal anomaly detection tasks compared with other state-of-the-art methods, the study selected methods from [9]-[16] (TSMAE, ITRAD, AEC, DTAC, M-iForest, MFS-CRNN, DSTR, CatBoost-DT) as comparison methods. These methods encompassed state-of-the-art (SOTA) technical approaches such as reconstruction models, ensemble learning, sequence modeling, and deep forests, representing the current optimal methods for multimodal detection. To establish a unified data benchmark, the study adopted the UNSW-NB15 benchmark dataset, which includes real-world IoT anomaly annotations, as a unified testing platform for heterogeneous device networks. It contained 49-dimensional features (protocol type/service type/connection status, etc.) and annotations for nine types of attack behaviors, integrating 2.5 million mixed protocol

(Modbus/TCP+HTTP) traffic records generated by real IoT devices. The experimental results are shown in Table 6.

As shown in Table 6, in the comprehensive performance evaluation, KM-A-E achieved the optimal anomaly detection rate of 96.2 ± 0.3% (95% CI [95.9%, 96.5%]), significantly outperforming CatBoost-DT (92.1% ± 0.4%, t = 14.37, p < 0.001). with a false positive rate as low as 1.8 ± 0.1% (95% CI [1.7%, 1.9%]), a 66.7% decrease compared to M-iForest (t=22.15, p<0.001). KM-A achieved the lowest inference latency of 8.3ms, making it the optimal solution for real-time performance (4.6% lower than M-iForest, t=3.18, p=0.012). At the deployment level, KM-A maintained the lowest model size of 29.7 MB (22.7% lower than DSTR, t=7.21, p<0.001) and the lowest energy consumption of 0.20 J/sample (11.1% lower than M-iForest, t=5.43, p=0.002), while achieving 0.99 protocol compatibility.

4.3 Parameter sensitivity verification

In this section, the study verified the sensitivity of the main parameters involved in the model. First was the edge point removal threshold. To verify the validity of the study setting for 3σ, the study preset different threshold gradients for verification. The experimental results are shown in Table 7.

In Table 7, in the edge point removal threshold sensitivity experiment (σ gradient: σ / 3σ / 5σ), when 3σ was used, the accuracy reached 0.89 ± 0.02 (95% CI [0.86, 0.92]), and the recall rate was 0.87 ± 0.03 (95% CI [0.83, 0.91]), significantly outperforming 5σ's recall rate of 0.71 (p=0.008, t=-4.32). The t-test showed that 3σ only increased the delay by 2.6 ms compared to σ (p=0.13, t=1.58), but memory usage was optimized by 5.3% (3σ 51.3 MB, σ 54.6 MB, p=0.02), and the training time remained at 48.7 ± 1.2 s. Subsequently, the study validated the effectiveness of the clustering process parameters, including the partitioning threshold α, the proportion of data anomalies γ, and the initial number of clusters k_{init}, with the results shown in Table 8.

Table 7: Edge point removal threshold sensitivity verification.

Parameter category	Parameters	Gradient value	Accuracy	Recall rate	Delay (ms)	Memory (MB)	Training time (s)
Edge point removal threshold	σ multiplier	σ	0.92	0.92	35.2	48.7	42.1
		3σ	0.89	0.87	37.8	51.3	48.7
		5σ	0.85	0.71	39.5	54.6	52.3

Table 8: Sensitivity verification of clustering process parameters.

Parameter category	Parameters	Gradient value	Accuracy	Recall rate	Delay (ms)	Memory (MB)	Training time (s)
Clustering process parameters	α	0.001	0.89	0.82	34.7	49.1	45.3
		0.01	0.91	0.91	36.5	50.5	49.6
		0.1	0.90	0.85	38.2	52.8	53.7
	γ	0.3	0.92	0.88	35.9	49.8	46.5
		0.5	0.91	0.91	36.2	50.2	49.1
		0.7	0.87	0.84	37.6	51.7	52.8
	k_{init}	$\sqrt{n}$	0.88	0.83	34.1	48.3	42.7
		$2\sqrt{n}$	0.91	0.91	36.0	50.9	49.8
		$4\sqrt{n}$	0.90	0.89	40.3	62.4	68.2

Table 9: DeepESN hyperparameter sensitivity verification.

Parameter category	Parameters	Gradient value	Accuracy	Recall rate	Delay (ms)	Memory (MB)	Training time (s)
DeepESN hyperparameters	N	60	0.89	0.84	21.3	38.5	35.1
		120	0.93	0.88	35.1	67.2	48.2
		240	0.94	0.85	72.6	128.9	83.7
	a	0.1	0.85	0.79	33.2	64.1	41.3
		0.3	0.93	0.88	35.1	67.2	48.5
		0.6	0.91	0.86	37.5	68.9	52.4
	L	1	0.89	0.85	28.7	51.3	39.6
		3	0.93	0.88	35.1	67.2	48.7
		5	0.92	0.87	53.4	105.6	72.3

In Table 8, in the clustering parameter sensitivity experiment, when α = 0.01, the accuracy reached 0.91 ± 0.01 (95% CI [0.89, 0.93]), which was significantly higher than when α = 0.1 by 1.1% (t = 2.87, p = 0.032). When γ = 0.5, the accuracy remained at 0.91 while reducing memory usage to 50.2 ± 0.8 MB, with no significant difference compared to γ = 0.3 (49.8 MB) (t = 1.03, p = 0.32). When k_{init} = $2\sqrt{n}$, the recall rate was 0.91 ± 0.02 (CI [0.88, 0.94]), an improvement of 9.7% compared to k_{init} = $\sqrt{n}$ (0.83) (t = 4.15, p = 0.004), with only a 1.9 ms increase in latency (p = 0.28). Finally, the effectiveness of the number of reserve pool layers L, the number of

neurons per layer N, and the leakage integral rate a in the DeepESN hyperparameters was verified, and the results are shown in Table 9.

In Table 9, in the DeepESN hyperparameter experiments, when N = 120, the accuracy reached 0.93 ± 0.01 (95% CI [0.91, 0.95]), a significant improvement of 4.5% compared to N = 60 (t = 8.14, p < 0.001), but the latency increased by 13.8 ms (t=5.22, p=0.002). a =0.3 improved the recall rate by 11.4% (a =0.30.88, a =0.10.79, t=7.33, p<0.001) compared to a =0.1 while maintaining an accuracy of 0.93 (CI [0.91, 0.95]). L =3 layers reduced training time by 32.9% (L =348.7 s, L = 572.3 s, t=9.06, p<0.001) and improved accuracy by 1.1% (L =30.88, L = 50.79, t=7.33, p<0.001) compared to L = 5 layers reduced training time by 32.9% (L =348.7s, L = 572.3s, t=9.06, p<0.001) and improved accuracy by 1.1% (L =30.93, L = 5 0.92), with memory remaining stable at 67.2±2.4MB (p>0.05).

5 Discussion and conclusion

In response to the problems of low efficiency and poor real-time performance of traditional IoT abnormal behavior detection models, the KM-A clustering algorithm was proposed and combined with the ESN algorithm to finally propose an IoT abnormal behavior detection model based on KM-A-E. The model improves the accuracy and real-time performance of abnormal behavior detection by optimizing the clustering performance of K-mean on datasets of different shapes and increasing the temporal weights of data features. The experimental results showed that in the simulation experiment, the detection accuracy of KM-A and KM-A-E in the manual dataset was between 0.91-0.99. The detection accuracy range of other algorithms was 0.69-0.91. After reducing the maximum number of iterations to 60, the recall rates of KM-A and KM-A-E for abnormal behavior detection ranged from 80.86% to 93.27%. The recall rate of other algorithms was between 60.05% and 77.78%. In the public dataset, the delay time of KM-A and KM-A-E was between 120ms-258ms. At this point, the delay of other algorithms was between 354ms and 1153ms. In practical application testing experiments, the detection models S-KA and S-KAE, with KM-A and KM-A-E as the core, achieved detection accuracies of 92.42% and 96.59%, respectively, for "回" shaped data. The detection accuracy of model S0 with K-means as the core was 74.62%. When detecting abnormal behavior in the local IoT, the detection coefficient error of S-KA and S-KAE was between 0-26. In ROC analysis, the AUC values of S-KA and S-KAE were 0.75 and 0.83, respectively.

At this point, the AUC value of S0 was 0.66. Compared with state-of-the-art methods, the KM-A-E and KM-A latency (181–258 ms) outperformed DSTR (260 ms in [15]), attributed to the direct mapping mechanism of the reserve pool in DeepESN eliminating gradient iteration calculations and SORM orthogonalization reducing matrix operations to O (1) complexity. However, this comes at the cost of introducing decision boundary blurring and hardware pre-configuration dependencies. KM-A-E achieved a 96.59% accuracy, significantly outperforming TSMAE (85% in Reference [9]), due to the dynamic fusion of geometric features through AGNES hierarchical clustering, and approached M-iForest (97.2% in [13]) with a <0.61% accuracy gap in exchange for a 23-fold delay compression (Pareto frontier validation confirms this trade-off) [25]. Its cross-scenario generalization capability benefited from DeepESN's adaptive adjustment of the leakage integral rate to data drift. In summary, research has practical application value in improving the accuracy and real-time performance of abnormal behavior detection in the IoT. However, the research model is insufficiently sensitive to contextual anomalies in domain semantic interpretation (such as multi-step collaborative attacks) due to the lack of behavioral logic association modeling in the current feature space, which does not cover network topology-level semantic reasoning. To address this issue, future research will focus on constructing a knowledge graph-driven semantic engine: integrating device metadata and threat intelligence, analyzing behavioral logic through spatiotemporal rule chains, strengthening cross-domain anomaly reasoning capabilities, and achieving protocol-level attack detection.

6 Funding

The research is supported by Research on Performance Evaluation of Undergraduate Education in Henan Province Based on Big Data Analysis, No.: 2021JKZD05; The 2021 Henan Province Undergraduate University Smart Teaching Special Research Project: Research and Practice of Personalized Teaching System in Local Comprehensive Universities in the Era of Smart Education.

References

[1] Mohsen Soori, Behrooz Arezoo, and Roza Dastres. Internet of things for smart factories in industry 4.0, a review. Internet of Things and Cyber-Physical Systems, 3(1):192-204, 2023.https://doi.org/10.1016/j.iotcps.2023.04.006

[2] Lijun Wu, Shidong Chen, and Zhicong Chen. Abnormal behavior detection based on attention-generative adversarial network. Microelectronics & Computer, 39(8):31-38, 2022.https://doi.org/10.19304/J.ISSN1000-7180.2022.0065

[3] Arnaldo Sgueglia, Andrea Di Sorbo, Corrado Aaron Visaggio, and Gerardo Canfora. A systematic literature review of IoT time series anomaly detection solutions. Future Generation Computer Systems, 134(1):170-186, 2022.https://doi.org/10.1016/j.future.2022.04.005

[4] Junaid Haseeb, Masood Mansoori, Yuichi Hirose, Harith Al-Sahaf, and Ian Welch. Autoencoder-based feature construction for IoT attacks clustering.

Future Generation Computer Systems, 127(1):487-502, 2022.https://doi.org/10.1016/j.future.2021.09.025

[5] S. K. Lakshminarayana, and P. I. Basarkod. Unification of K-Nearest Neighbor (KNN) with distance aware algorithm for intrusion detection in evolving networks like IoT. Wireless Personal Communications, 132(3):2255-2281, 2023.https://doi.org/10.1007/s11277-023-10722-8

[6] Dominik Olszewski, Marcin Iwanowski, and Waldemar Graniszewski. Dimensionality reduction for detection of anomalies in the iot traffic data. Future Generation Computer Systems, 151(1):137-151, 2024.https://doi.org/10.1016/j.future.2023.09.033

[7] Sergio Trilles, Sahibzada Saadoon Hammad, and Ditsuhi Iskandaryan. Anomaly detection based on artificial intelligence of things: A systematic literature mapping. Internet of Things, 25:101063, 2024.https://doi.org/10.1016/j.iot.2024.101063

[8] Beibei Li, Shang Ma, Ruilong Deng, Kim-Kwang Raymond Choo, and Jin Yang. Federated anomaly detection on system logs for the internet of things: A customizable and communication-efficient approach. IEEE Transactions on Network and Service Management, 19(2):1705-1716, 2022.https://doi.org/10.1109/TNSM.2022.3152620

[9] Honghao Gao, Binyang Qiu, Ramón J. Durán Barroso, Walayat Hussain, Yueshen Xu, and Xinheng Wang. Tsmae: A novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder. IEEE Transactions on Network Science and Engineering, 10(5):2978-2990, 2022.https://doi.org/10.1109/TNSE.2022.3163144

[10] Gen Li, and Jason J. Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. Information Fusion, 91(1):93-102, 2023.https://doi.org/10.1016/j.inffus.2022.10.008

[11] Hao Xu, Zihan Sun, Yuan Cao, and Hazrat Bilal. A data-driven approach for intrusion and anomaly detection using automated machine learning for the internet of things. Soft Computing, 27(19):14469-14481, 2023.https://doi.org/10.1007/s00500-023-09037-4

[12] Alessandra De Benedictis, Francesco Flammini, Nicola Mazzocca, Alessandra Somma, and Francesco Vitale. Digital twins for anomaly detection in the industrial internet of things: Conceptual architecture and proof-of-concept. IEEE Transactions on Industrial Informatics, 19(12):11553-11563, 2023.https://doi.org/10.1109/TII.2023.3246983

[13] Orieb AbuAlghanam, Hadeel Alazzam, Esra'a Alhenawi, Mohammad Qatawneh, and Omar Adwan. Fusion-based anomaly detection system using modified isolation forest for internet of things. Journal of Ambient Intelligence and Humanized Computing, 14(1):131-145, 2023.https://doi.org/10.1007/s12652-022-04393-9

[14] Nenavath Chander, and Mummadi Upendra Kumar. Metaheuristic feature selection with deep learning enabled cascaded recurrent neural network for anomaly detection in industrial internet of things environment. Cluster Computing, 26(3):1801-1819, 2023.https://doi.org/10.1007/s10586-022-03719-8

[15] Wenbin Huang, Kun Xie, and Jie Li. A novel sequence tensor recovery algorithm for quick and accurate anomaly detection. IEEE Transactions on Network Science and Engineering, 9(5):3531-3545, 2022.https://doi.org/10.1109/TNSE.2022.3189365

[16] Maryam Douiba, Said Benkirane, Azidine Guezzaz, and Mourade Azrour. An improved anomaly detection model for IoT security using decision tree and gradient boosting. The Journal of Supercomputing, 79(3):3392-3411, 2023.https://doi.org/10.1007/s11227-022-04783-y

[17] Sristi Vashisth, and Anjali Goyal. Dynamic anomaly detection using robust random cut forests in resource-constrained IoT environments. Informatica, 48(23):107-120, 2024.https://doi.org/10.31449/inf.v48i23.6862

[18] Zhenpeng Zhang. SD-WSN network security detection methods for online network education. Informatica, 48(21):51-66, 2024.https://doi.org/10.31449/inf.v48i21.6257

[19] Manoj Kumar Gupta, and Pravin Chandra. Effects of similarity/distance metrics on k-means algorithm with respect to its applications in IoT and multimedia: A review. Multimedia Tools and Applications, 81(26):37007-37032, 2022.https://doi.org/10.1007/s11042-021-11255-7

[20] K. Ramesh Chandra, and Somasekhar Borugadda. Energy efficiency enhancement in millimetre-wave MIMO-NOMA using three-layer user grouping and adaptive power allocation algorithm. Sustainable Computing: Informatics and Systems, 43:100991, 2024.https://doi.org/10.1016/j.suscom.2024.100991

[21] Jiaming Jiang, Guoheng Ruan, and Zhenggan Dai. Retrieval technology of enterprise data center resources based on density peak clustering algorithm. Computing and Informatics, 42(4):923-942, 2023.https://doi.org/10.31577/cai_2023_4_923

[22] Martin Higgins, Bruce Stephen, and David Wallom. Detecting smart meter false data attacks using hierarchical feature clustering and incentive weighted anomaly detection. IET Cyber-Physical Systems: Theory & Applications, 8(4):257-271, 2023.https://doi.org/10.1049/cps2.12057

[23] Wenqi Qiu, Wu Ai, Huazhou Chen, Quanxi Feng, and Guoqiang Tang. Decentralized federated learning for industrial IoT with deep echo state networks. IEEE Transactions on Industrial Informatics, 19(4):5849-5857, 2022.https://doi.org/10.1109/TII.2022.3194627

[24] Wen Zhang, Mimi Xie, Caleb Scott, and Chen Pan. Sparsity-aware intelligent spatiotemporal data sensing for energy harvesting IoT system. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 41(11):4492-4503, 2022.https://doi.org/10.1109/TCAD.2022.3197543

[25] Chaofan Hou, Nan Xu, and Siyu Liu. Design of online monitoring method for distribution iot devices based on DBSCAN optimization algorithm. Informatica, 49(5):181-194, 2025.https://doi.org/10.31449/inf.v49i5.6399